

Button Group

Button Group Component This document outlines the specifications for the `ButtonGroupComponent`, an Angular component designed to group multiple buttons together, providing a cohesive interface for selecting options or triggering actions.

Purpose

- The `ButtonGroupComponent` facilitates the organization of multiple buttons into a single cohesive unit, allowing users to select options or trigger actions efficiently. It enhances user experience by visually grouping related buttons and providing a clear indication of available choices or actions.

Usage

- Integrate the `ButtonGroupComponent` into your Angular applications to create visually unified groups of buttons for various purposes, such as selecting filters, navigating between views, or triggering actions.

Inputs

- `buttons` : (Required) An array of objects representing the buttons within the button group. Each object should have the following properties:
 - `label` : (Required) The label or display text for the button.
 - `value` : (Required) The value associated with the button.
 - `disabled` : (Optional) Disables the button if set to true.
 - `icon` : (Optional) The icon to be displayed alongside the button label.
 - `selectedValue` : (Optional) The value of the initially selected button within the group.
- `layout` : (Optional) Specifies the layout orientation of the button group. Possible values: 'horizontal' : Arranges buttons horizontally (default). 'vertical' : Arranges buttons vertically.

Events

- `selectionChange` : Fired when the selected button within the group changes. Emits the value of the newly selected button.

```
< app-button-group [buttons] = "buttonGroupItems" [selectedValue] = "selectedButtonValue" layout = "horizontal" (selectionChange) = "onButtonGroupSelectionChange($event)" > </ app-button-group >
```

Notes

- Customize the styling and layout of the button group component to align with the overall design language and branding of your application. Implement the `selectionChange` event handler to respond to changes in the selected button within the group. Ensure that each button in the `buttons` array has a unique `value` property to distinguish between buttons and facilitate selection tracking.

```
// Example buttons for button group  
buttonGroupItems = [ { label : 'Option 1' , value : 'option1' } , { label : 'Option 2' , value : 'option2' } , { label : 'Option 3' , value : 'option3' , disabled : true } , { label : 'Option 4' , value : 'option4' , icon : 'fa fa-check' } ] ; // Example selected button value  
selectedButtonValue : string = 'option1' ; onButtonGroupSelectionChange ( selectedValue : string ) { // Handle button group selection change event }
```

```
<app-button-group
  [buttons]="buttonGroupItems"
  [selectedValue]="selectedButtonValue"
  layout="horizontal"
  (selectionChange)="onButtonGroupSelectionChange($event)"
></app-button-group>

---

// Example buttons for button group
buttonGroupItems = [
  { label: 'Option 1', value: 'option1' },
  { label: 'Option 2', value: 'option2' },
  { label: 'Option 3', value: 'option3', disabled: true },
  { label: 'Option 4', value: 'option4', icon: 'fa fa-check' }
];

// Example selected button value
selectedButtonValue: string = 'option1';

onButtonGroupSelectionChange(selectedValue: string) {
  // Handle button group selection change event
}
```